Optics Communications 285 (2012) 594–608

Contents lists available at SciVerse ScienceDirect

Optics Communications

journal homepage: www.elsevier.com/locate/optcom

Image encryption using P-Fibonacci transform and decomposition

Yicong Zhou [a,*], Karen Panetta [b], Sos Agaian [c], C.L. Philip Chen [a]

[a] Department of Computer and Information Science, University of Macau, Macau, China
[b] Department of Electrical and Computer Engineering, Tufts University, Medford, MA USA 02155
[c] Department of Electrical and Computer Engineering, University of Texas at San Antonio, San Antonio, TX USA 78249

ARTICLE INFO

Article history:
Received 27 September 2011
Received in revised form 12 November 2011
Accepted 15 November 2011
Available online 28 November 2011

Keywords:
P-Fibonacci transform
Fibonacci p-code bit-plane decomposition
Image encryption
Privacy protection

ABSTRACT

Image encryption is an effective method to protect images or videos by transferring them into unrecognizable formats for different security purposes. To improve the security level of bit-plane decomposition based encryption approaches, this paper introduces a new image encryption algorithm by using a combination of parametric bit-plane decomposition along with bit-plane shuffling and resizing, pixel scrambling and data mapping. The algorithm utilizes the Fibonacci P-code for image bit-plane decomposition and the 2D P-Fibonacci transform for image encryption because they are parameter dependent. Any new or existing method can be used for shuffling the order of the bit-planes. Simulation analysis and comparisons are provided to demonstrate the algorithm's performance for image encryption. Security analysis shows the algorithm's ability against several common attacks. The algorithm can be used to encrypt images, biometrics and videos.

1. Introduction

With the explosive growth in wired and wireless digital communication and ubiquitous internet multimedia services, enormous and diverse technologies are available to individuals all over the world to create, distribute, and access images and videos. Providing security for these images and videos containing proprietary or private information becomes an important issue for individuals, business and governments. Image encryption is an effective approach [1] to protect images or videos by transforming them into unrecognizable formats such that unauthorized users have difficulty decoding the encrypted objects. Examples of the many applications requiring robust security methods includes preserving privacy for medical images in clinical applications, enforcing copyright protection for design graphs, images and videos for commercial purposes, as well as providing security for personal identification via fingerprinting or iris matching and for video monitoring in homeland security applications.

Images or videos can be partially or fully encrypted by using different technologies in the spatial domain or the frequency domain. Image/video encryption in the frequency domain is mainly based on the Discrete Cosine Transform (DCT) [2], Fresnel Transform (FrT) [3,4] or Fractional Fourier Transform (FrFT) [5–7]. These algorithms attempt to scramble or encrypt the transform coefficients or blocks.

Image/video encryption in the spatial domain can protect images or videos with a desired level of security while providing a high level of quality. Encryption algorithms in the spatial domain are based on scrambling image/video pixels or blocks using different technologies. One straightforward method is the naïve encryption algorithm [8]. This scheme considers the image or video as a data sequence or stream. It scrambles or encrypts part or the entire sequence or data stream using different techniques. Data Encryption Standard (DES) [9] and Advanced Encryption Standard (AES) [10] are two examples of this method. Nevertheless, this method requires significant computational resources [11] and has the worst error resilience performance [12].

Many encryption schemes are based on chaos theory since the chaotic maps or systems can generate random noise-like sequences iteratively for given initial conditions and parameters [13–18]. However, their resulting sequences are real numbers which need to transform into integer or binary sequences according to additional conditions or thresholds for data encryption purposes, requiring extra computation cost.

Recursive sequences have been applied to image encryption recently because they directly generate integer sequences for specific parameters or keys. These recursive sequences include the Fibonacci numbers [19,20]. However, these approaches provide a low level of security due to the lack of security keys or the small key space. To improve the efficiency of the encryption process and the security level of the encrypted objects, our preliminary work resulted in several image encryption algorithms using different recursive sequences such as the P-Fibonacci sequence [21] and P-recursive sequence [22]. These algorithms are only permutation based methods which are known to be vulnerable for plaintext attacks [23]. To achieve higher levels of security, an effective solution is to change image pixel values while scrambling image pixels or blocks using different techniques [24].

Another interesting encryption approach in the spatial domain is based on image bit-plane decomposition. This method first decomposes images into their binary bit-planes. It then encrypts bit-planes using

* Corresponding author. Tel.: +853 83978458; fax: +853 28838314.
E-mail address: yicongzhou@umac.mo (Y. Zhou).

0030-4018/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.optcom.2011.11.044

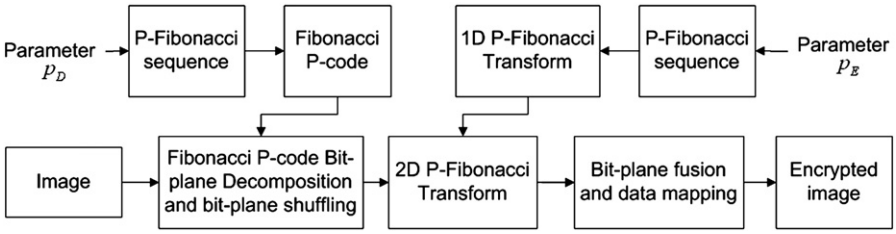

Fig. 1. The basic flow graph of the new encryption algorithm where P_D is the decomposition parameter and P_E is the parameter for encryption.

different technologies, and combines all the encrypted bit-planes to obtain the encrypted images. Recently, a bit-plane encryption algorithm using exclusive-OR operations (BPE-XOR) was presented for an optical system [25]. Later, a selective bit-plane encryption scheme using the AES algorithm (SBE-AES) was developed for image encryption in mobile environments [26]. To reduce the computational workload, another selective bit-plane encryption algorithm using the least significant bit-plane of images (SBE-LBP) was proposed to encrypt images [27].

These bit-plane decomposition based encryption schemes have made contributions for their specific applications. However, they have some security weaknesses: (1) they are based on traditional image bit-plane decomposition. From a cryptanalysis point of view, this decomposition method has low level of security because the number of its decomposed bit-planes and the content of each bit-plane for a specific image are fixed. For example, a grayscale image with gray levels 0–255 can only be decomposed into eight bit-planes. The contents in each bit-plane are fixed for the same image. As a result, the attacker can easy to predict the decomposed results. (2) The XOR operation and selective bit-plane encryption schemes have been shown to be vulnerable to a low computational cost attack [28].

Benefited from the recursive sequence based encryption algorithms, this paper introduces a new image encryption algorithm in order to enhance the security of the bit-plane decomposition based encryption approaches. The new algorithm combines the Fibonacci P-code bit-plane decomposition with the newly introduced P-Fibonacci transforms. It decomposes the input images into the Fibonacci P-code bit-planes, shuffles the order of bit-planes, resizes the bit-planes based on the 2D P-Fibonacci transforms, encrypts all bit-planes one by one using the 2D Fibonacci transform, combines all encrypted bit-planes, and then maps the image data back into the original data range of the input images to obtain the final resulting encrypted images. The deep analysis and comparison demonstrate the new algorithm's encryption performance. A comprehensive cryptanalysis is provided to address the algorithm's security issues. The basic flow graph of the algorithm is shown in Fig. 1.

The rest of this paper is organized as follows: Section 2 will introduce two P-Fibonacci transforms which will be used for the new encryption algorithm presented in Section 4. Section 3 will present the Fibonacci P-code bit-plane decomposition. Section 5 will provide several simulation examples to show the performance of the new algorithm for image encryption. A cryptanalysis of the encryption algorithm will be addressed in Section 6. Conclusions will be drawn in Section 7.

2. P-Fibonacci transforms

In this section, we introduce two P-Fibonacci transforms. One is the 1D P-Fibonacci transform which is used to generate a permutation sequence of an input sequence. The other is the 2D P-Fibonacci transform which encrypts 2D/3D images by scrambling image pixel locations.

2.1. P-Fibonacci sequence

Here, we review the definitions and some properties of the P-Fibonacci sequence [29].

Definition 2.1. The P-Fibonacci sequence is a recursive sequence defined by,

$$F_p(i) = \begin{cases} 0 & i<0 \\ 1 & i=0 \\ F_p(i-1)+F_p(i-p-1) & i>0 \end{cases} \tag{1}$$

where i is a position index of the sequence and a non-negative integer p is a distance parameter.

Based on the definition in Eq. (1), the P-Fibonacci sequence changes with different p values. Some examples are given in Table 1. Note that the P-Fibonacci sequence can derive the power of two series ($p=0$) and traditional Fibonacci number ($p=1$).

Here is a list of properties of the P-Fibonacci sequence [30].

$$\sum_{i=0}^{n-1} F_p(i) = F_p(n+p)-1 \tag{2}$$

$$\sum_{i=n-p}^{n-1} F_p(i) = F_p(n+p), \quad n \geq p \tag{3}$$

$$F_p(n) = \binom{p}{0} + \binom{n-p}{1} + \binom{n-2p}{2} + \ldots + \binom{m+r}{m} \tag{4}$$

where $n=(p+1)m+r$

2.2. 1D P-Fibonacci transform

Definition 2.2. Let $f_p(i)$ and $f_p(i+1)$ be two consecutive elements of the P-Fibonacci sequence defined in Eq. (1). The following transformation is called the 1D P-Fibonacci transform.

$$\begin{pmatrix} T_1 \\ T_2 \\ \ldots \\ T_N \end{pmatrix} = \left(f_p(i)+\varepsilon\right) \begin{pmatrix} 1 \\ 2 \\ \ldots \\ N \end{pmatrix} \bmod f_p(i+1) \tag{5}$$

where $f_p(i)+\varepsilon<f_p(i+1)$, $N=f_p(i+1)-1$, the non-negative integer i is the index location of the P-Fibonacci sequence. The constant ε is a minimal integer offset such that the greatest common divisor of $f_p(i)+\varepsilon$ and $f_p(i+1)$ is one.

Table 1
P-Fibonacci sequences with different p values.

P	i									
	0	1	2	3	4	5	6	7	8	–
0	1	2	4	8	16	32	64	128	256	–
1	1	1	2	3	5	8	13	21	34	–
2	1	1	1	2	3	4	6	9	13	–
3	1	1	1	1	2	3	4	5	7	–
4	1	1	1	1	1	2	3	4	5	–
–	–									
∞	1	1	1	1	1	1	1	1	1	–

Both constraints are important for this transform. The first one $f_p(i)+\varepsilon<f_p(i+1)$ is a limitation for choosing the minimal offset ε. The second one $N=f_p(i+1)-1$ specifies the maximum value of the input sequence. Otherwise, the input sequence has to be resized to meet this condition. For example, we assume $p=2$ and the input sequence in Eq. (5) is (1, 2, 3, 4, 5, 6, 7, 8, 9, 10), thus $N=10$. Applying $p=2$ to Eq. (1) generates a P-Fibonacci sequence 1, 1, 1, 2, 3, 4, 6, 9, 13, 19…, then select $f_2(i+1)=13$ and $f_2(i)=9$. To meet the second constraint in Eq. (5), namely, $N=f_p(i+1)-1=12$, the input sequence has to be resized to (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12). Using Eq. (5) generates an output sequence (9, 5, 1, 10, 6, 2, 11, 7, 3, 12, 8, 4), which is a permutation of the input sequence.

Moreover, the 1D P-Fibonacci transform is periodic since it contains a modulo operation. The periodic property of this type of matrix transforms was discussed mathematically [31].

Based on Eq. (5), a sequence (1, 2, …, N) can be transformed into its permutation sequence (T_1, T_2, …, T_N). The output of the 1D P-Fibonacci transform changes with different parameter p values. For instance, if the input sequence is (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12), the output sequence will be (8, 3, 11, 6, 1, 9, 4, 12, 7, 2, 10, 5) for $p=1$, and (9, 5, 1, 10, 6, 2, 11, 7, 3, 12, 8, 4) for $p=2$; if the input sequence is (1, 2, 3, 4, 5, 6, 7, 8), the output is (5, 1, 6, 2, 7, 3, 8, 4) for $p=2$.

2.3. 2D P-Fibonacci transforms

Definition 2.3. Let A be a 2D image with size $M\times N$, C_r and C_c be the row and column coefficient matrices respectively. The following transformation is called the 2D P-Fibonacci transform:

$$E = C_r A C_c \tag{6}$$

where E is the encrypted image and

$$C_r(u,v)=\begin{cases}1 & \text{for}(u,T_u)\\ 0 & else\end{cases}\ \text{and}\ C_c(x,y)=\begin{cases}1 & \text{for}\left(T_y,y\right)\\ 0 & else\end{cases}$$

where T_u and T_y are obtained from Eq. (5), and $1\le u, v\le M$, $1\le x, y\le N$.

Additionally, the 1D P-Fibonacci transform is a special case of the 2D P-Fibonacci transform. When A is a 1D matrix, $A=(1, 2, \ldots, N)^T$, the 1D P-Fibonacci transform can be represented in another format,

$$\begin{pmatrix}T_1\\T_2\\\cdots\\T_N\end{pmatrix}=C_r\begin{pmatrix}1\\2\\\cdots\\N\end{pmatrix} \tag{7}$$

where C_r is the row coefficient matrix defined in Eq. (6).

The 2D P-Fibonacci transform can be used to encrypt the 2D and 3D images such as grayscale images, biometrics, color images and medical images. Based on Definition 2.3, to encrypt an $M\times N$ 2D image, the row coefficient matrix C_r is an $M\times M$ matrix, and the column coefficient matrix C_c is an $N\times N$ matrix. Note that the original image should be resized such that the row and column sizes meet the Fibonacci transform to generate the permutated row and column sequences and then their corresponding coefficient matrices.

For example, for an input image with size 8×12 and $p=2$, the row and column coefficient matrices of the 2D P-Fibonacci transform will be,

$$C_r=\begin{pmatrix}0&0&0&0&1&0&0&0\\1&0&0&0&0&0&0&0\\0&0&0&0&0&1&0&0\\0&1&0&0&0&0&0&0\\0&0&0&0&0&0&1&0\\0&0&1&0&0&0&0&0\\0&0&0&0&0&0&0&1\\0&0&0&1&0&0&0&0\end{pmatrix}\quad C_c=\begin{pmatrix}0&0&1&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&1&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&1&0&0&0\\0&0&0&0&0&0&0&0&0&0&0&0&1\\0&1&0&0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&1&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&1&0&0&0&0\\0&0&0&0&0&0&0&0&0&0&0&1&0\\1&0&0&0&0&0&0&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&1&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0&1&0&0\end{pmatrix}$$

Definition 2.4. Let E be a 2D encrypted image, C_r^{-1} and C_c^{-1} be inverse matrices of C_r and C_c defined in Definition 2.3 respectively. The following transformation is called the inverse 2D P-Fibonacci transform:

$$R = C_r^{-1} E C_c^{-1} \tag{8}$$

where R is the reconstructed image.

Similarly, the inverse 1D P-Fibonacci transform can be also defined by,

$$\begin{pmatrix}1\\2\\\cdots\\N\end{pmatrix}=C_r^{-1}\begin{pmatrix}T_1\\T_2\\\cdots\\T_N\end{pmatrix} \tag{9}$$

This give a simple and effective way to reconstruct the original input sequence for the 1D P-Fibonacci transform.

In image decryption process, the user simply applies the inverse 2D P-Fibonacci transform one time to reconstruct the original images.

3. Fibonacci P-code bit-plane decomposition

In this section, we review the Fibonacci P-code bit-plane decomposition. This paper will investigate its application in image encryption. The term “grayscale image” refers to grayscale image with gray levels within 0–255 in the rest of this paper.

3.1. Fibonacci P-code

Definition 3.1. The non-negative decimal number D can be represented by the following form of the base-2 polynomial.

$$D=\sum_{i=0}^{n-1}a_i 2^i = a_0 2^0 + a_1 2^1 + \ldots + a_{n-1}2^{n-1} \tag{10}$$

The binary code (a_{n-1}, …, a_1, a_0) is the binary representation of the non-negative decimal number D.

A non-negative decimal number can be represented by a binary code sequence based on Eq. (10). This concept can be extended to the Fibonacci P-code since the power of two series is a special case of the P-Fibonacci sequence. Therefore, the definition of Fibonacci P-code is given below.

Definition 3.2. A non-negative decimal number D can be represented by the following format:

$$D=\sum_{i=0}^{n-1}c_i f_p(i) = c_0 f_p(0) + c_1 f_p(1) + \ldots + c_{n-1} f_p(n-1) \tag{11}$$

Table 2
Different Fibonacci p-codes of 30 for p = 3.

$f_3(i)$	26	19	14	10	7	5	4	3	2	1	1	1
Fibonacci p-codes	1	0	0	0	0	0	1	0	0	0	0	0
	1	0	0	0	0	0	0	1	0	1	0	0
	0	1	0	1	0	0	0	0	0	1	0	0
	0	1	0	0	1	0	1	0	0	0	0	0
	–											

where n and p are non-negative integers, $i = 0, 1, \ldots, n-1$, $c_i \in (0, 1)$, $f_p(i)$ is the i^{th} element of the P-Fibonacci sequence with a specific p value in Eq. (1). The coefficient sequence $(c_{n-1}, \ldots, c_1, c_0)$ is called the Fibonacci P-code of D, namely,

$$D = (c_{n-1}, \ldots, c_1, c_0)_p \tag{12}$$

where p is the distance parameter of the P-Fibonacci sequence in Eq. (1).

The Fibonacci P-code of a specific decimal number will change for different p values. This is because the P-Fibonacci sequence will be different when the p value changes. However, for a given p value, the Fibonacci P-code of a specific decimal number is not unique either.

For example, if $D = 30$ and $p = 3$, i.e. $D = (30)_{p=3}$, using Eq. (1) would yield a P-Fibonacci sequence with 12 elements. Each element serves as a weight in the Fibonacci P-code. The decimal number 30 can be represented by different Fibonacci P-codes with 12 bits as shown in Table 2.

In order to obtain a unique Fibonacci P-code for each non-negative decimal number, several different rules or constraints were presented [29,32,33]. The users have flexibility to choose one of them. In this paper, we select the constraints presented in [29], namely

$$D = f_p(i) + s \tag{13}$$

Where the $f_p(i)$ is the i^{th} element of the P-Fibonacci sequence with a specific p value in Eq. (1), $0 \le i < n$, and a non-negative decimal number s is a reminder, $0 \le s < f_p(i-p)$.

Based on the constraints in Eq. (13), for $D = 30$ and $p = 3$, its Fibonacci P-code is $30 = (1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0)_3$. Note that there are at least p 0's between two consecutive 1's in the Fibonacci P-code of any non-negative decimal number after applying the constraints in Eq. (13).

3.2. Fibonacci P-code bit-plane decomposition

A grayscale image can be decomposed into eight 1-bit binary bit-planes [34]. This traditional bit-plane decomposition is widely used in image compression and enhancement. However, its decomposition results and the number of bit-planes are unchangeable for a specific grayscale image and easy to predict. This is not conducive for image encryption.

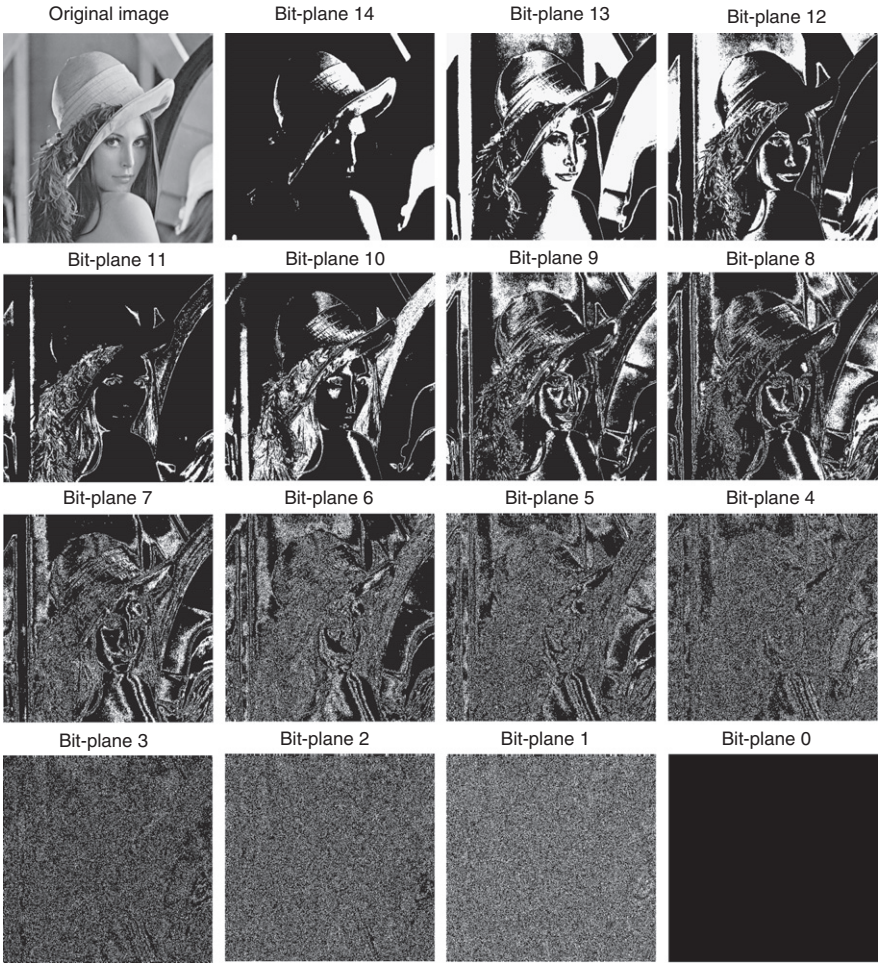

Fig. 2. Fibonacci P-code bit-plane decomposition of the grayscale Lena image, p = 2.

For a given p value, each non-negative decimal number has a unique Fibonacci P-code. With the same concept of the traditional bit-plane decomposition, an image can be also decomposed into several Fibonacci P-code bit-planes. Moreover, the traditional bit-plane decomposition is a special case of the Fibonacci p-code bit-plane decomposition because the P-Fibonacci sequence becomes the power of two series when $p=0$.

The number of Fibonacci P-code bit-planes n_B depends on the maximum value of images $I_{\max}$. In order to have the decomposition method works over all p values, we introduce the following rules. If $p \le I_{\max}$, n_B is calculated from $I_{\max}$; Otherwise, if $p > I_{\max}$, n_B is calculated by p values. This means that for the latter case, the number of Fibonacci P-code bit-planes is only determined by the p value.

For a specific grayscale image, the results of the Fibonacci P-code bit-plane decomposition are parameter dependent. The number of the Fibonacci P-code bit-planes n_B changes with parameter p values. For instance, for a grayscale image, $I_{\max}=255$. Therefore, for $p=2$, $n_B=15$, and for $p=3$, $n_B=19$. Fig. 2 gives an example of a grayscale image decomposed using the Fibonacci P-code bit-plane decomposition with $p=2$.

Moreover, the contents of the Fibonacci P-code bit-planes are different based on different p values. These advantages make the Fibonacci P-code bit-plane decomposition well suitable for image encryption. Note that the maximum value of medical images could be greater than 255 so the presented decomposition approach will also work for other types of images.

4. New image encryption algorithm

In this section, we introduce a new image encryption algorithm using the P-Fibonacci transforms and Fibonacci P-code bit-plane decomposition, called the P-Fibonacci Encryption (PFE) algorithm. It can be used to encrypt images biometrics, and videos.

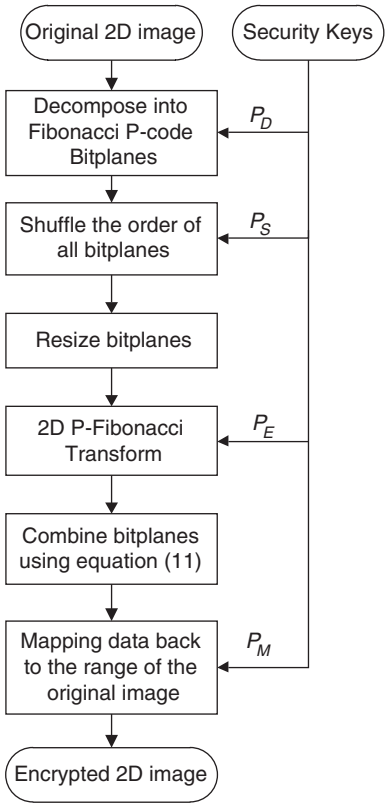

Fig. 3. The block diagram of the new PFE algorithm.

The new PFE algorithm shown in Fig. 3 contains five processes: image decomposition, bit-plane shuffling, bit-plane resizing, bit-plane encryption and data mapping processes. The algorithm decomposes the original image into its Fibonacci P-code bit-planes, shuffles the order of all bit-planes, resizes bit-planes based on the size of 2D P-Fibonacci transforms, encrypts the all bit-planes one by one using 2D P-Fibonacci transform, combines all encrypted bit-planes using binary code definition in Eq. (10) and then map the image data back into the original image data range to generate the final resulting encrypted image.

Let $\{X_0, X_1, \ldots, X_{L-1}\}, X_0 < X_1 < \ldots < X_{L-1}$, denote all discrete intensity levels in an input image $I(m,n)$, the data mapping function is defined by,

$$E(m,n) = k \quad \text{for } I(m,n) = X_k \tag{14}$$

where $E(m,n)$ is the output encrypted image, $k=0, 1, \ldots, L-1$.

The Fibonacci P-code and P-Fibonacci transform will be different when parameter p changes. Both image decomposition and encryption processes are parameter dependent. The parameter p for both decomposition process (called P_D) and encryption process (called P_E) can act as security keys for the new PFE algorithm. The users have flexibility to choose the same p value for both processes, i.e. $P_D=P_E$, or select the different p values for each process, namely $P_D \neq P_E$. They can also select different P_E values for each bit-plane which helps to achieve a higher level of security but also increases computational complexity.

The image resizing process helps to improve the security level of the PFE algorithm because it makes it difficult for an attacker to decode the encrypted images. The users have flexibility to choose any existing method to perform the shuffling process. The data mapping process changes and spreads image data while keeping the encrypted images within the same data range of the original images.

Except for image resizing process, other four processes in the new PFE algorithm are parameter dependent. Its security keys consist of the parameters in these four processes, namely (1) P_D for image decomposition, (2) security key P_S for bit-plane shuffling, (3) P_E for bit-plane encryption, and (4) the pixel value array P_M for data mapping.

To recover the original image from the encrypted image, the authorized users will be provided the combination of the security keys and the image resizing method. The decryption process of the new PFE algorithm first maps encrypted image data back into its original range, and then decomposes the image into its binary bit-planes which are the Fibonacci P-code bit-planes generated in the encryption process, reverts the order of all bit-planes back to their original order, decrypts all bit-planes one by one using the 2D Fibonacci transform, resizes all bit-planes back to their original, and combines all decrypted bit-planes to obtain the reconstructed image.

5. Experimental results

The PFE algorithm has been successfully applied to more than fifty images, including grayscale images, biometrics, color images, and medical images such as Magnetic Resonance Images (MRIs) and Computer Tomography (CT) images. To show encryption performance of the PFE algorithm, we provide several illustrative examples of image encryption in this section.

In all simulation results obtained by the PFE algorithm in the rest of this paper, we use the same security key P_E for all bit-planes and simply reverse the order of the Fibonacci P-code bit-planes in the shuffling process. The random numbers are added in the padding region in the image resizing process. Of course, the users have flexibility to use other methods to shuffle the order of the bit-planes and to resize the original images.

The structural similarity (SSIM) index is a quantitative assessment method for measuring the similarity between two images [35]. The SSIM is used to quantitatively evaluate the similarity between the reconstructed and original images to demonstrate whether the original images are completely reconstructed or not. A value 1 of the SSIM index indicates that two measured images are identical. The Matlab code of the SSIM is obtained from the author webpage [36]. We simply use its default setting to measure the images.

Fig. 4 gives an illustrative example of a grayscale image encrypted by the PFE algorithm with large values of the security keys, $P_D = 32$ and $P_E = 300$. The results show that images changes in different encryption and decryption stages. The encrypted image shown in Fig. 4(d) is visually different from the original image shown in Fig. 4(a). The original images are completely reconstructed because the reconstructed image shown in Fig. 4(f) is visually the same as the original one and the SSIM value 1 confirms that the two of them are identical.

Fig. 5 provides four encryption results to show that the new PFE algorithm has capability of protecting different types of images such as grayscale image, MR images, CT images and fingerprints. All these images are encrypted by the PFE algorithm with $P_D = 10$ and $P_E = 15$. The encrypted images are visually close to noise images. These show the excellent encryption performance of the PFE algorithm. All the reconstructed images and their SSIM=1 demonstrate that the original images are perfectly reconstructed.

The 3D images such as color images and 3D medical images contain several 2D data matrices called 2D components. Color images, for example, contain three color planes. Each color plane is a 2D component. In this manner, the 3D images can be considered as the combination of several 2D images. The 3D image encryption can be accomplished by encrypting all its 2D components one by one. The users have flexibility to choose the same security keys for all 2D components or different security keys for each of them.

Fig. 6 gives an descriptive example of color image encryption using the PFE algorithm with the security keys: $P_D = 10$ and $P_E = 20$. The encrypted image shown in Fig. 6(d) visually looks like a noise image. The results also demonstrate the PFE algorithm's capability of encrypting 3D images. The reconstructed image shown in Fig. 6(f) and its SSIM value further verify that the original image is also completely reconstructed.

6. Security analysis

Security is important not only for the encrypted objects but also for the encryption algorithms themselves. In this section, we discuss some security issues of the new PFE algorithm such as histogram and correlation analysis, key sensitivity test, security key space as well as several common attacks such as brute force attacks, noise attacks, and data loss attacks. Eight images with different sizes shown in Fig. 7 will be used as test images in this section.

6.1. Histogram analysis

An image histogram is a graphic representation of the pixel intensity distribution of an image. To overcome statistic attacks, the encrypted image should have a histogram with random behavior and uniform distribution [37,38].

Fig. 8 shows an example of image encryption using the new PFE algorithm. The encrypted image and its histogram are completely different from the original image. The encrypted image visually looks like a noise image. Its histogram has nearly uniform distribution

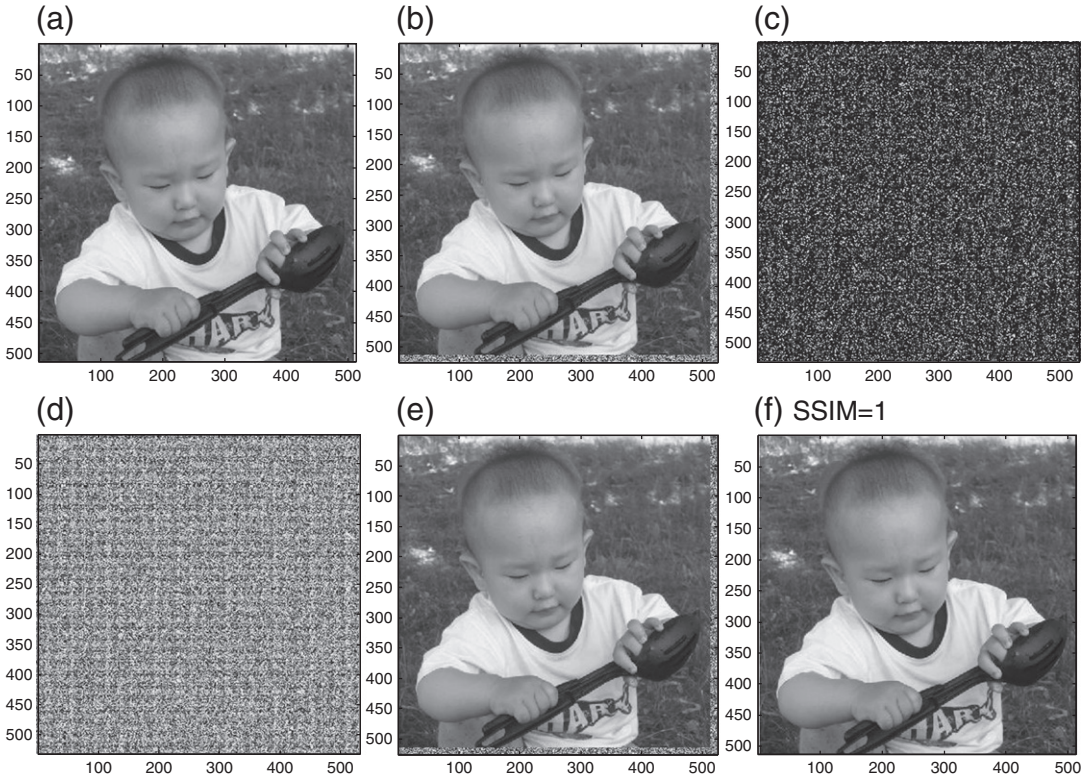

Fig. 4. Grayscale image encryption using the PFE algorithm, $P_D = 32$ and $P_E = 300$. (a) The original image; (b) the resized image; (c) the encrypted image before the data mapping; (d) the final encrypted image; (e) the reconstructed image without image resizing; (f) the final reconstructed image.

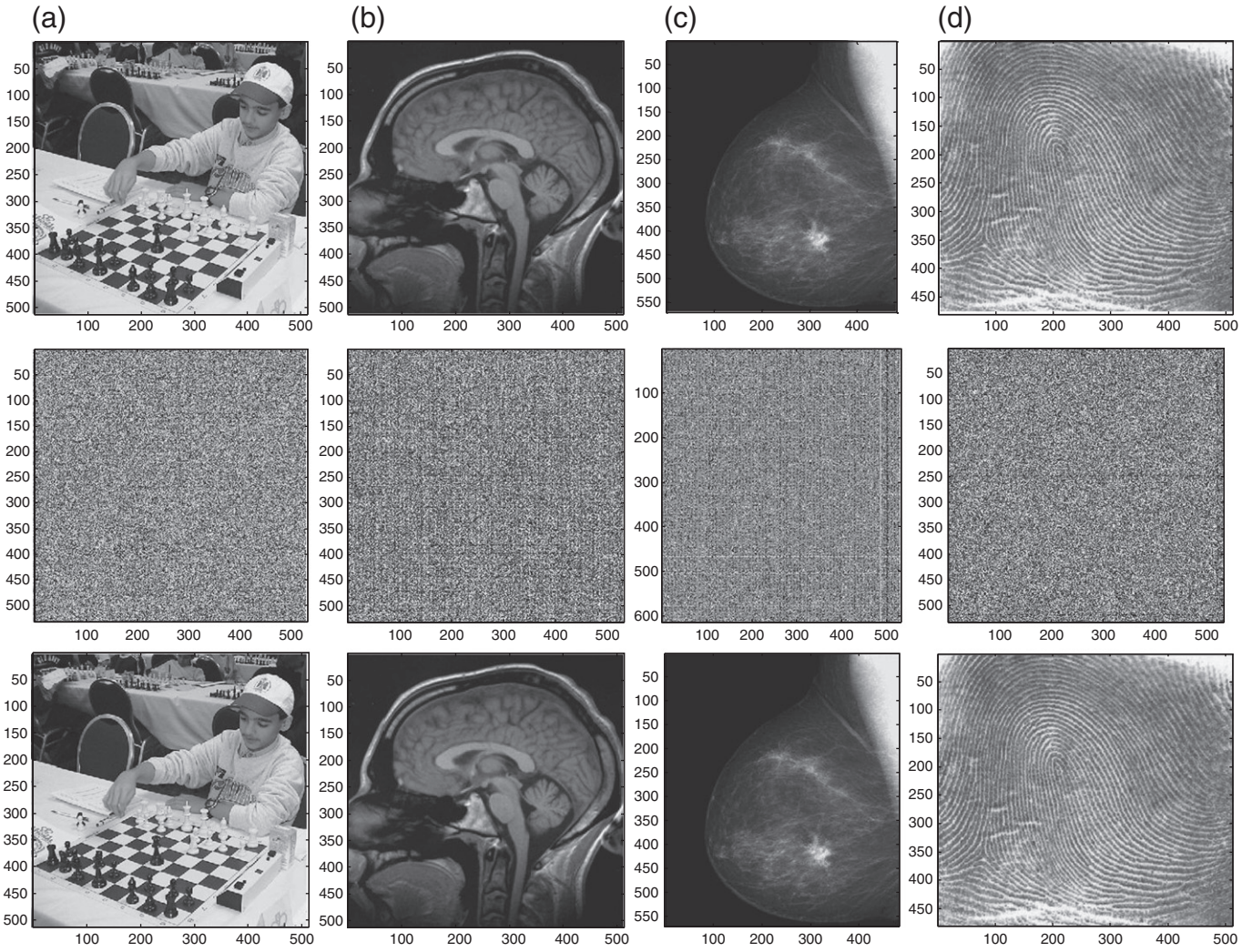

Fig. 5. Encryption for different types of images, $P_D = 10$ and $P_E = 15$. The first row shows the original images; the second row shows the encrypted images; the third row shows the reconstructed images. (a) The grayscale image case; (b) the MRI case; (c) the CT image case; (d) the biometrics case. This demonstrates that the PFE algorithm has capability to encrypt different types of images and the original images can be completely reconstructed.

which changes with different security keys. These demonstrate that the PFE algorithm has capability of withstanding the statistic attacks.

Moreover, the reconstructed image shown in Fig. 8(c) is visually the same as its original one in Fig. 8(a). Its SSIM value also shows that they are identical. To further quantitatively and graphically show the difference between the reconstructed and original images, a difference image is generated by subtracting the reconstructed image from the original image pixel by pixel. The histogram of the difference image is shown in Fig. 8(c). The result shows that all pixels in this image are zeros. This further demonstrates that the reconstructed image is the same as the original image. This is one of advantages of the presented PFE algorithm.

To further verify the encryption performance of the new PFE algorithm, we compare it with those previously mentioned existing algorithms such as the bit-plane encryption algorithm using exclusive-OR operations (BPE-XOR) [25], the selective bit-plane encryption algorithm using the AES algorithm (SBE-AES) [26], and the selective bit-plane encryption algorithm using the least significant bit-plane of images (SBE-LBP) [27]. Four images selected from Fig. 7 are encrypted by these algorithms, respectively. The encrypted images and their histograms are shown in Fig. 9. The resulting images encrypted by the BPE-XOR, SBE-AES and SBE-LBP algorithms contain some visual information of the original images. The intensity distribution of their corresponding histograms is inhomogeneous. However, the encrypted images by the PFE algorithm visually look like noise images. The histograms of these encrypted images are close to uniform distribution. This demonstrates that the PFE algorithm shows better performance than other methods in against statistic attacks.

6.2. Correlation coefficient analysis

To further show the new PFE algorithm to be robust against statistic attacks, we analyze the correlation between two horizontally, vertically and diagonally neighboring pixels in the original and encrypted images. The neighboring pixels are also called adjacent pixels [37,38].

First, we study the intensity distribution of two neighboring pixels in the original and encrypted images by the PFE algorithm. 2048 sample pixels are randomly selected from the original image shown in Fig. 8(a) and the encrypted images shown in Fig. 8(b), respectively.

Fig. 10 plots the intensity distribution of these 2048 sample pixels and their horizontally, vertically and diagonally neighboring pixels. The top row in Fig. 10 shows the intensity distributions of pixels from the original image. The results show that intensity values of

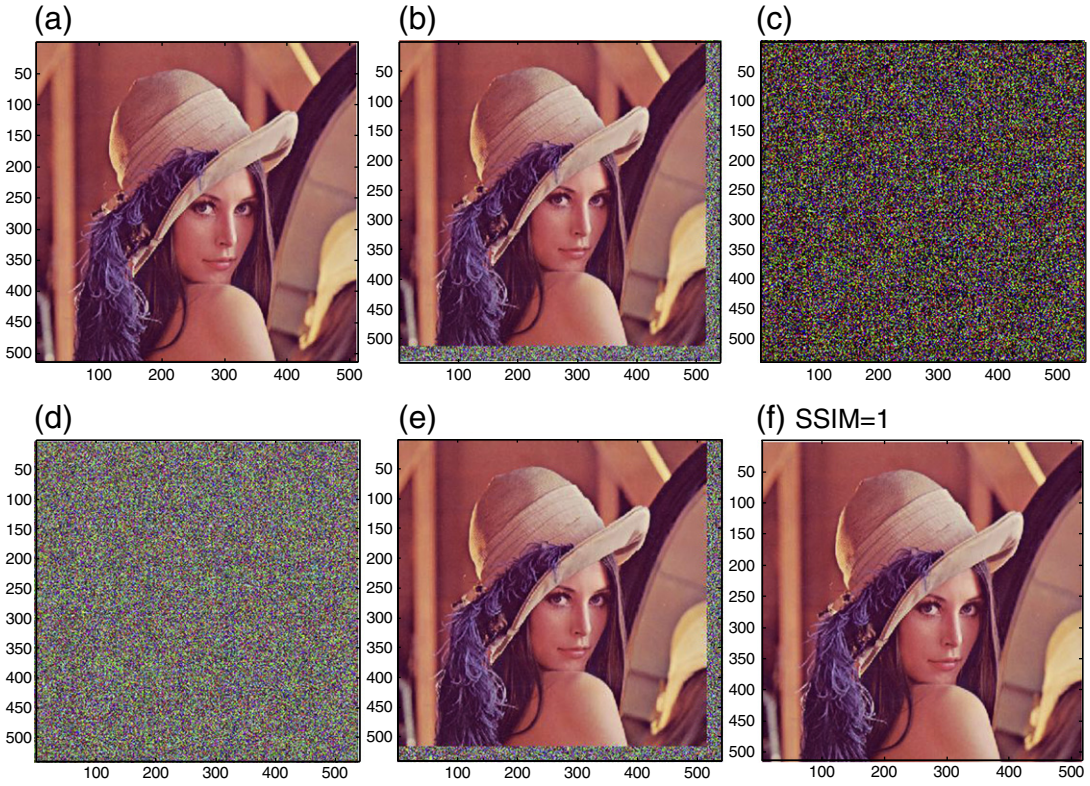

Fig. 6. Color image encryption using the PFE algorithm, $P_D = 10$ and $P_E = 20$. (a) The original color image; (b) the resized color image; (c) the processed image before the data mapping process; (d) the encrypted image; (e) the reconstructed image without image resizing; (f) the final reconstructed image.

neighboring pixels are equal or very close because their intensity distribution is located in or close to the diagonal line in the figures. The neighboring pixels in the original image are highly correlated. The bottom row in Fig. 10 plots the distribution of pixels from the encrypted image. The results, however, demonstrate that the neighboring pixels in the encrypted image show less correlation because their intensity values spread out and nearly uniformly distributed in the entire data range of the image.

To quantitatively assess the correlation of the neighboring pixels, we calculate the correlation coefficient of all horizontally, vertically

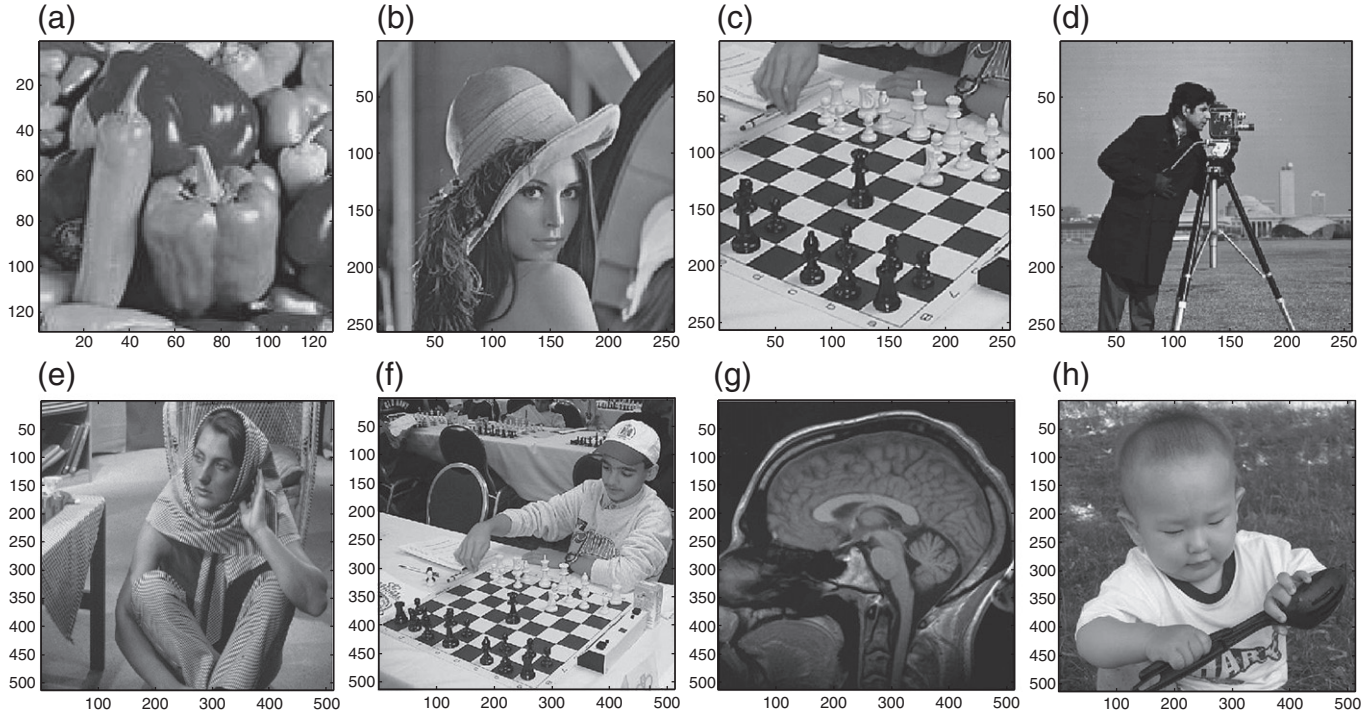

Fig. 7. Test images with different sizes. (a) Pepper, 128×128; (b) Lena, 256×256; (c) Chess, 256×256; (d) Cameraman, 256×256; (e) Barbara, 512×512; (f) Chess player, 512×512; (g) Brain, 512×512; (h) Baby, 512×512.

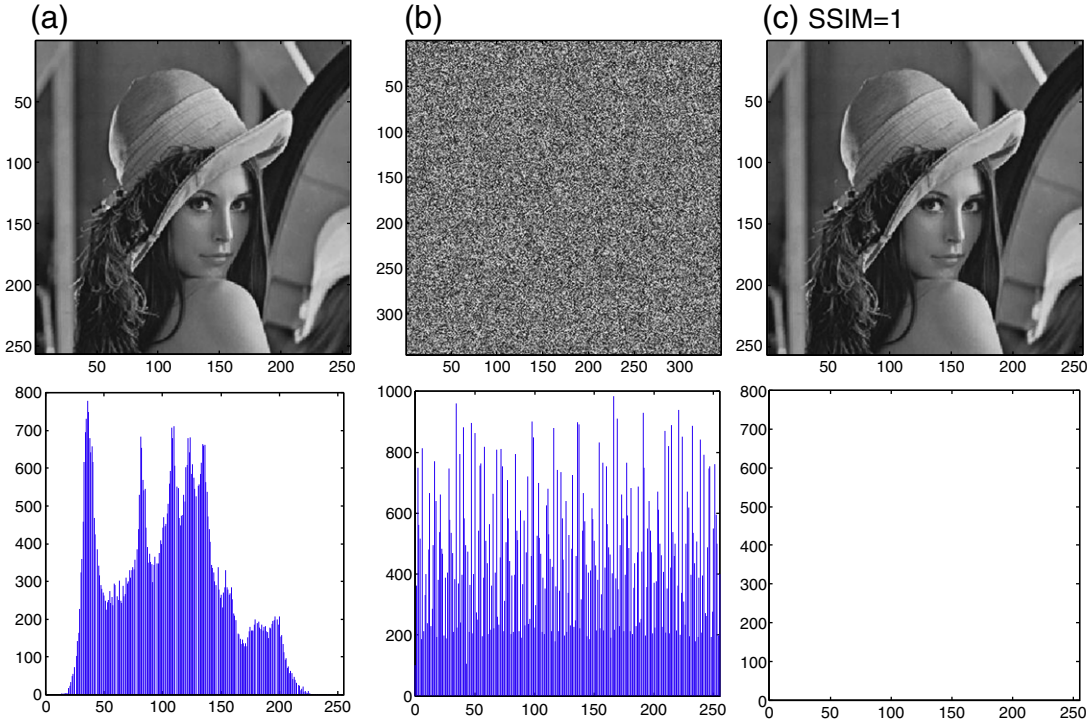

Fig. 8. Image encryption using the PFE algorithm. (a) The original image and its histogram; (b) the encrypted image and its histogram; (c) the reconstructed image and the histogram of the difference between the reconstructed and original images.

and diagonally neighboring pixels in the original and encryption images. The correlation coefficient of two neighboring pixels is defined by [39,40],

$$r_{xy} = \frac{N\sum_{i=1}^{N} x_i y_i - \sum_{i=1}^{N} x_i \sum_{i=1}^{N} y_i}{\sqrt{\left(N\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2\right)\left(N\sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} y_i\right)^2\right)}} \tag{15}$$

where x, y are intensity value of two neighboring pixels in image and N is the total number of pixels selected from the image to be calculated, and $-1 \le r_{xy} \le 1$.

Two neighboring pixels x, y have a strong positive linear correlation if the correlation coefficient r_{xy} is close to $+1$. The positive values of the correlation coefficient r_{xy} indicate a relationship between two neighboring pixels x, y such that the values of x increase (decrease) while the values of y increase (decrease). However, a value of r_{xy} close to zero implies that there is a random, nonlinear relationship between the two neighboring pixels [40].

Table 3 shows the correlation coefficients of two horizontally, vertically and diagonally neighboring pixels in the original images shown in Fig. 7 and the encrypted images by the PFE algorithm. The correlation coefficients of two neighboring pixels in the original images are close to one. This shows that they have a strong positive relationship. However, the correlation coefficients of two neighboring pixels in the encrypted images are close to zeros which means they have an extremely weak relationship.

To evaluate the relationship between the original images and their corresponding encrypted images by the presented PFE algorithm, we calculate the correlation coefficients of two pixels with the same locations in the original and encrypted images. They are shown in the last column in Table 3. The results demonstrate that no linear correlation occurs between the original images and the encrypted image since the values of the correlation coefficients are close to zeros.

Table 4 compares the average values of the correlation coefficients of two neighboring pixels in three directions within the original images and the encrypted images by four different algorithms. The results of all original images are close to one. The correlation coefficients of the encrypted images are close to zero. The PFE algorithm outperforms other encryption algorithms because the correlation coefficients of its encrypted images are closer to zeros. This is further confirmed by the average values of all original images and encrypted images by each algorithm in the bottom row. This comparison further demonstrates that a strong relationship is presented in the neighboring pixels in the original image and a very week correlation is shown in the neighboring pixels in the encrypted images by the new PFE algorithm.

6.3. Key sensitivity test

The security keys of the PFE algorithm are the combination of (1) P_D for image decomposition, (2) security key for bit-plane shuffling, (3) P_E for bit-plane encryption, and (4) the pixel value array for data mapping. These security keys are very important for the algorithm. The users have flexibility to choose same or different security keys for both decomposition and encryption processes. The number of the Fibonacci P-code bit-planes in the image decomposition process depends on the length of Fibonacci P-code which differs based on different P_D values.

An ideal encryption algorithm should be sensitive with the security key change [38]. A small change of the security key should result in a completely different encrypted image and vice versa. To test the key sensitivity of the PFE algorithm, we try to use different security keys to reconstruct the original image. An example is shown in Fig. 11. A Chess player image shown in Fig. 7(f) is encrypted by the new PFE algorithm with selecting $P_D = 3$ for the decomposition process and $P_E = 3$ for the encryption process, and reversing the order of the bit-planes for bit-plane shuffling. We try to reconstruct the original image by using different P_D for decomposition process but keeping the security key P_E the same as that for encryption process. The results shown in Fig. 11 verify that the original image can be completely

reconstructed only when the correct security keys are being utilized. Otherwise, the reconstructed images are completely different with the original one even if P_D for the image reconstruction is slightly different with that for image encryption as examples in Fig. 11(c) and (d). This demonstrates that the PFE algorithm is highly sensitive with the security key changes.

6.4. Security key space

We use an $M \times N$ image as an example to calculate the security key space for the PFE algorithm. We assume that the security key P_D for the decomposition process has K_D possible choices. The number of the decomposed Fibonacci P-code bit-planes is n_B for a specific P_D. Since any new or existing method can be used for the bit-plane shuffling process, the maximum possible changes of bit-planes are $n_B!$ (the factorial of n_B). We also assume that the possible choices of P_E for each bit-plane are $K_E (K_E \leq M!N!)$. The pixel value array for a specific image in the data mapping process is determined by the image decomposition and bit-plane shuffling process. Therefore, the security key space for the presented PFE algorithm will be,

$$S = K_D n_B! (K_E)^{n_B} \leq K_D n_B! (M!N!)^{n_B} \tag{16}$$

Table 5 gives some examples of key space based on the assumption of $K_E = 10$, the specific P_D values ($K_D = 1$) and a 64×64 grayscale

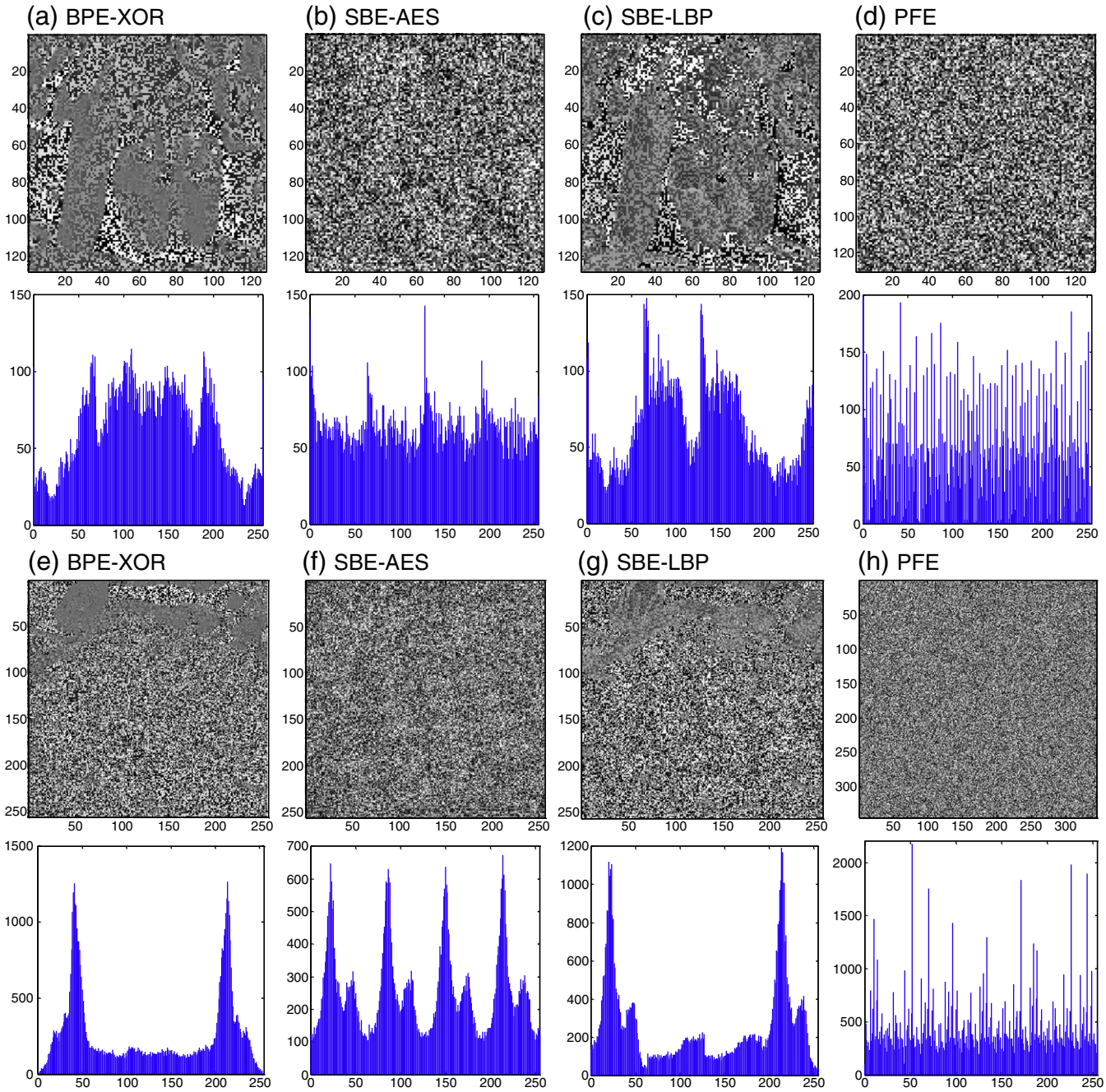

Fig. 9. Comparison the histograms of the encrypted images by different algorithms. The 1st column shows the encrypted images by the BPE-XOR and their histograms; the 2nd column shows the encrypted images by the SBE-AES and their histograms; the 3rd column shows the encrypted image by the SBE-LBP and their histograms; the 4th column shows the encrypted image by the PFE algorithm and their histograms. (a)–(d) shows the encrypted images of Pepper image shown in Fig. 7(a); (e)–(h) shows the encrypted images of Chess image shown in Fig. 7(c); (i)–(l) shows the encrypted images of Baby image shown in Fig. 7(h); (m)–(p) shows the encrypted images of Brain image shown in Fig. 7(g).

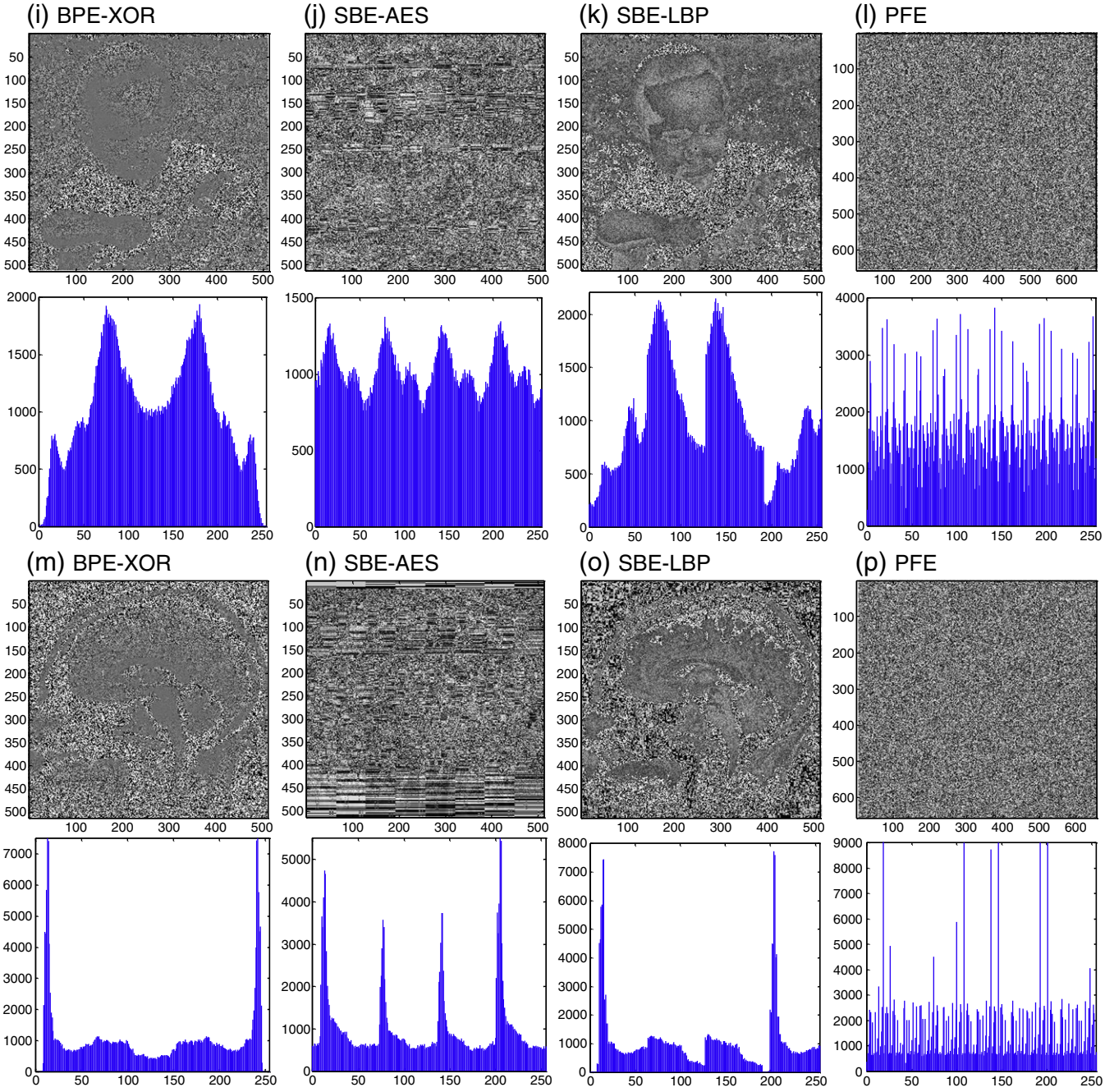

Fig. 9 (*continued*).

image. This shows that the security key space of the PFE algorithm is sufficient large.

6.5. Brute force attack

In cryptanalysis, the brute force attack [41] is an attack model in which the attacker tries to guess the security keys by performing an exhaustive search for all possibilities of the security keys of the encryption algorithm. Theoretically, this approach is feasible if the key space of the encryption algorithm is limited and the attacker knows the encryption algorithm.

Based on the results of Table 5, the security key space of the PFE algorithm is large enough even if we choose P_D to be a specific value and only 10 possible choices for P_E. As a result, the PFE algorithm can withstand the brute force attack.

6.6. Noise attacks

The communication and networking channels are generally in presence of different types of noise. To test the robustness of the new PFE algorithm against noise attacks, it is compared with other existing bit-plane decomposition based encryption methods. The original image shown in Fig. 7(f) is encrypted by these encryption algorithms individually. The Salt & Pepper noise with density 0.05 is added to the encrypted images. We then try to reconstruct the original image from these noised encrypted images. The SSIM index is used to quantitatively evaluate the similarity between the

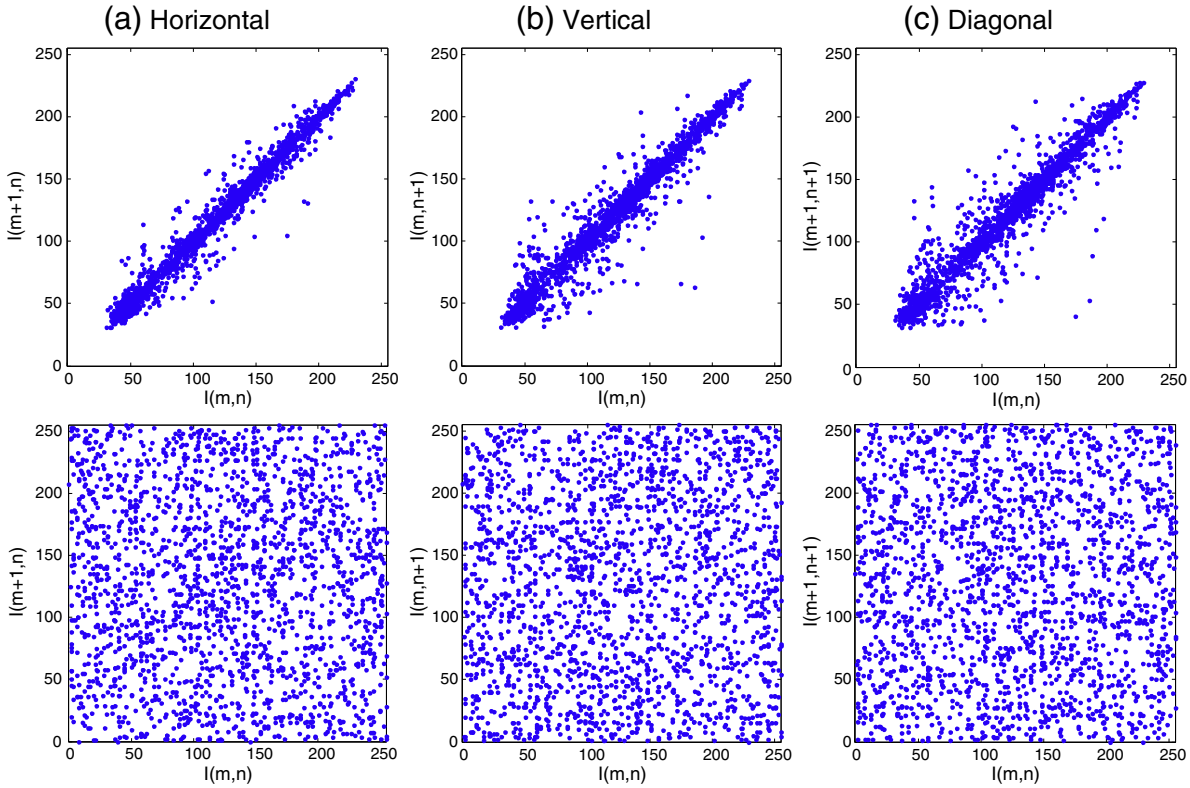

Fig. 10. Pixel intensity distributions of two neighboring pixels at different directions in the original and encrypted Lena image in Fig. 7. The top row shows the pixel intensity distributions of the original image; the bottom row shows the pixel intensity distributions of the encrypted image. (a) The pixel intensity distribution of two horizontally neighboring pixels, $I(m, n)$ and $I(m+1, n)$; (b) the pixel intensity distribution of two vertically neighboring pixels, $I(m, n)$ and $I(m, n+1)$; (c) the pixel intensity distribution of two diagonally neighboring pixels, $I(m, n)$ and $I(m+1, n+1)$. This demonstrates that the neighboring pixels at different directions in the encrypted image show less correlation and more random distribution in the entire data range.

reconstructed images and the original images. The results are shown in Fig. 12.

The original image cannot be reconstructed for the SBE-AES and SBE-LBP algorithms as shown in Fig. 12(b) and (c). The presented PFE algorithm and the BPE-XOR can reconstruct the original images although they are in presence of noise, as shown in Fig. 12(a) and (d). The SSIM result of the PFE algorithm is higher than that of the BPE-XOR. This demonstrates that the PFE algorithm show better performance than other methods for resisting the noise attacks.

6.7. Data loss attacks

Data loss attacks are to test the capability of the encryption algorithm for tolerating the data loss during the public media transmission channels.

The Lena image in Fig. 8(a) is encrypted by the PFE algorithm and other existing methods. The data within a 20×20 window in the center of the encrypted images are removed by replacing them as zeros. We then try to reconstruct the original image from these encrypted images with data loss. The reconstructed images are then evaluated by the SSIM index measure. The results are shown in Fig. 13.

All the reconstructed images shown at the bottom row in Fig. 13 contain most information of the original image. These algorithms show good performance for resisting the data loss attack. In the same condition of the data loss attack, the PFE algorithm can preserve more visual information compared to other methods. Its reconstructed image show more visually pleasing than others although its SSIM value is slightly lower than the BPE-XOR. This demonstrates that the presented PFE algorithm outperforms other methods in against data loss attacks.

Table 3
Correlation coefficients of two neighboring pixels in the original images in Fig. 7 and their encrypted images.

Image name	Original image			Encrypted image			Correlation between two images
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal	
Pepper	0.9147	0.9362	0.9351	−0.0036	−0.0016	0.0042	−0.0069
Lena	0.9401	0.9698	0.9699	−0.00087	0.0016	0.0028	−0.0019
Chess	0.9476	0.9071	0.9077	−0.00066	−0.0015	0.0028	0.000963
Cameraman	0.9343	0.9408	0.9412	0.0099	0.00081	0.005	0.000734
Barbara	0.8545	0.9539	0.9540	0.0043	0.0019	0.0045	0.000094
Chess player	0.9639	0.9475	0.9475	−0.00048	0.00053	0.0009	−0.0014
Brain	0.9866	0.9857	0.9857	−0.0013	−0.0011	0.001	−0.0033
Baby	0.9899	0.9846	0.9846	−0.00063	0.001	0.0022	0.000327

Table 4
Comparison of average correlation coefficients of two neighboring pixels within the original and encrypted images.

Image name	Original	BPE-XOR	SBE-AES	SBE-LBP	PFE
Pepper	0.9287	0.014825	0.03517	0.055918	−0.00153
Lena	0.9599	−0.00303	0.055022	0.050406	0.004461
Chess	0.9208	0.00216	0.026635	0.014548	0.001805
Cameraman	0.9388	0.001571	0.095544	0.093596	0.003426
Barbara	0.9208	−0.00307	0.024869	0.920733	0.00345
Chess player	0.9530	0.000844	0.040877	0.024003	0.000572
Brain	0.9860	−0.00063	0.205399	0.133599	0.000336
Baby	0.9864	0.001938	0.115748	0.053715	−0.0004
Average	0.9493	0.001826	0.074908	0.168315	0.001515

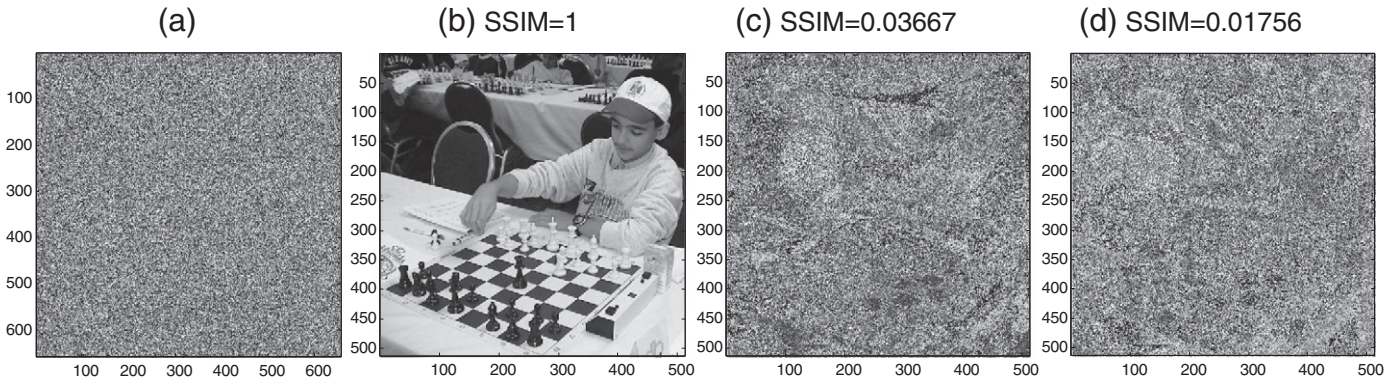

Fig. 11. Image reconstruction using the same $P_E = 3$for the encryption process but different P_D for the decomposition process. (a) Encrypted image, $P_D = 3$; (b) reconstructed image, $P_D = 3$; (c) reconstructed image , $P_D = 5$; (d) reconstructed image, $P_D = 0$. This demonstrates that the original image can be completely reconstructed only when the correct security keys are being utilized.

6.8. Plaintext attacks

There are two types of plaintext attacks: the known-plaintext attack and the chosen-plaintext attack [41,42]. In Chosen-plaintext attack, the attacker has the flexibility to choose any useful information as plaintext in order to deduce the security keys of the encryption algorithm, or reconstruct the original plaintexts from the unknown ciphertexts. If an encryption algorithm does not change the image data, the attacker has a high probability to partially or entirely break the encrypted images by using plaintext attacks without knowing the encryption algorithm and its security keys.

In the presented PFE algorithm, the image data have been changed by four steps: (1) shuffling the order of all bit-planes in shuffling process; (2) scrambling pixel positions in the encryption process if the P_E is different for each bit-plane; (3) Combining all encrypted bit-planes back to the gray levels using binary numeral system; (4) Mapping the encrypted image data back into the grayscale image data range [0, 255] by using a data mapping function in Eq. (14). Therefore, the PFE algorithm changes both the image pixel locations and intensity values. It has capability of withstanding plaintext attacks.

7. Conclusion

The goal of this paper is to introduce a new approach for improve the security level of bit-plane decomposition based encryption algorithms. We have introduced a new image encryption algorithm by combining two well-known approaches: image bit-plane decomposition and the image pixel permutation using recursive sequences. We have chosen the p-Fibonacci sequence as an example of recursive sequences and the Fibonacci P-code bit-plane decomposition as a decomposition case which includes the traditional image bit-plane decomposition (i.e. $P_D = 0$). The presented PFE algorithm, on the other hand, has demonstrated a new application of the Fibonacci P-code and its bit-plane decomposition for image encryption. The PFE algorithm consists of five processes: a decomposition process, a bit-plane shuffling process, a bit-plane resizing process, an encryption process and a data mapping process.

Both decomposition and encryption processes are parameter dependent. The security keys for both of them have a large number of possible combinations. Any new or existing method can be used for the bit-plane shuffling process. The algorithm changes image pixel positions in the encryption process while changing the image pixel values in the data mapping, bit-plane shuffling and resizing

Table 5
Security key spaces with different P_D values.

P_D	0	1	2	3	–
n_B	8	12	15	19	–
$n_B!$	8!	12!	15!	19!	–
K_E	10	10	10	10	–
S	4.03×10^{12}	4.79×10^{20}	1.31×10^{27}	1.22×10^{36}	–

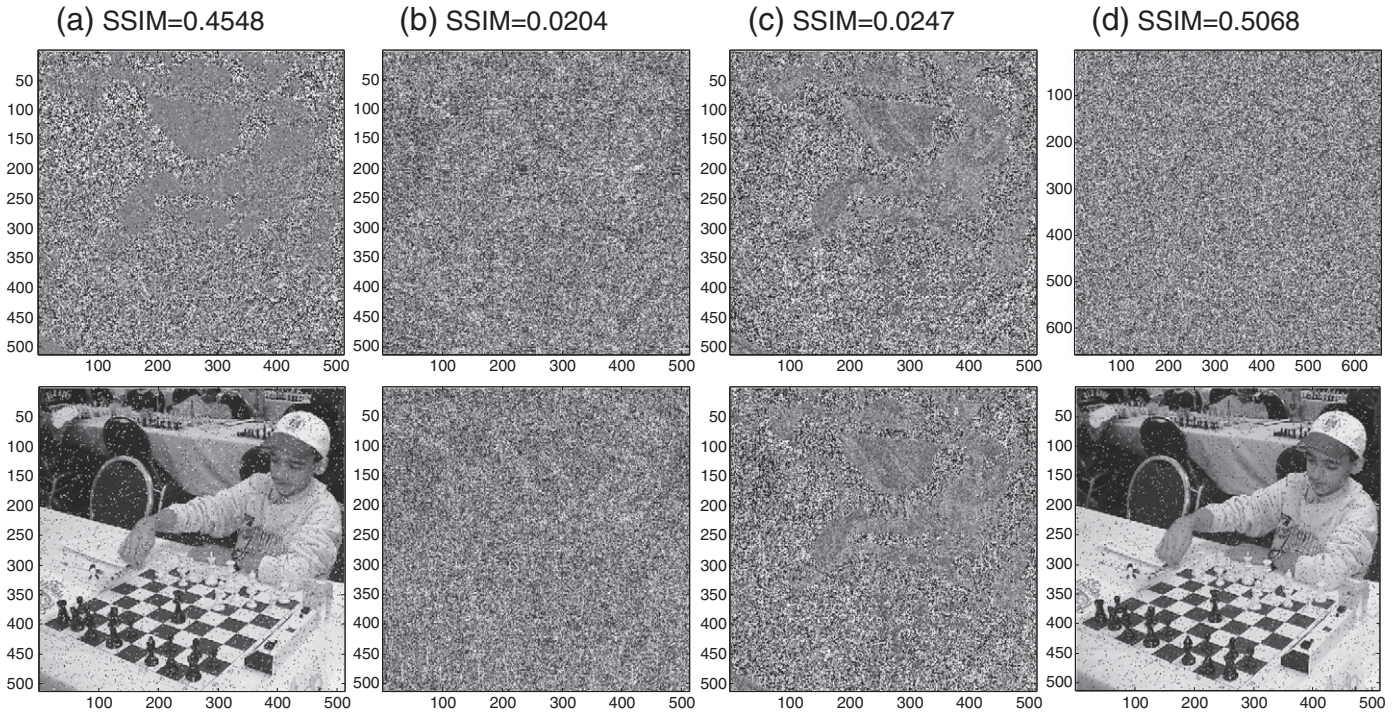

Fig. 12. Performance comparison of different algorithms against noise attacks. (a)–(d) shows the noised encrypted images by different algorithms and their corresponding reconstructed images. The top row shows the encrypted images with 0.05 Salt & Pepper noise added. The bottom row shows the reconstructed images from the corresponding noised encrypted images. (a) BPE-XOR; (b) SBE-AES; (c) SBE-LBP; (d) the PFE algorithm.

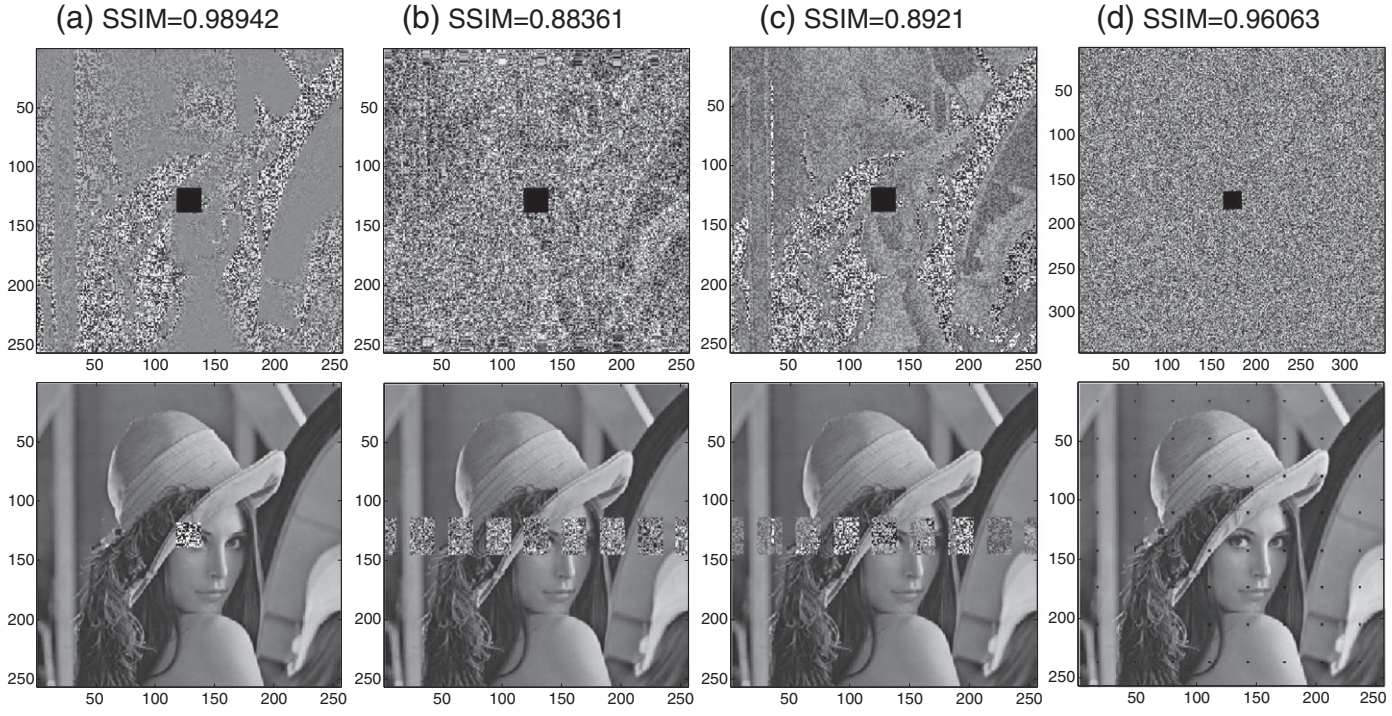

Fig. 13. Performance comparison of different algorithms against data loss attacks. (a)–(d) shows the encrypted images by different algorithms with data loss and the corresponding reconstructed images. The top row shows the encrypted images with data removal within a 20×20 center window. The bottom row shows the reconstructed images. (a) BPE-XOR; (b) SBE-AES; (c) SBE-LBP; (d) the PFE algorithm.

processes. All of these ensure unauthorized user's difficulty for decoding the protected images.

Finally, computer simulations have demonstrated the performance of the PFE algorithm. Security analysis has shown the capability of presented PFE algorithm for withstanding several common attacks such as the brute force, statistic, noise, data loss and plaintext attacks. The PFE algorithm can be used to encrypt images, biometrics and videos. Our future work will extend the PFE algorithm to the 3D.

References

[1] N. Zhou, Y. Wang, L. Gong, Optics Communications 284 (2011) 3234.
[2] Z. Liu, L. Xu, T. Liu, H. Chen, P. Li, C. Lin, S. Liu, Optics Communications 284 (2011) 123.
[3] H.T. Chang, H.-E. Hwang, C.-L. Lee, Optics Communications 284 (2011) 4146.
[4] H.-E. Hwang, Optics Communications 284 (2011) 3243.
[5] F. Ge, L.F. Chen, D.M. Zhao, Optics Communications 281 (2008) 4254.
[6] N. Zhou, T. Dong, J. Wu, Optics Communications 283 (2010) 3037.
[7] N. Zhou, Y. Wang, L. Gong, H. He, J. Wu, Optics Communications 284 (2011) 2789.
[8] S. Nandi, B.K. Kar, P. Pal Chaudhuri, IEEE Transactions on Computers 43 (1994) 1346.
[9] National Institute of Standards and Technology, Data Encryption Standard (DES), in: http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf (Ed.), 1999.
[10] J. Daemen, V. Rijmen, Proceedings of The International Conference on Smart Card Research and Applications, Springer-Verlag, 2000.
[11] M. Grangetto, E. Magli, G. Olmo, IEEE Transactions on Multimedia 8 (2006) 905.
[12] B.B. Zhu, M.D. Swanson, S. Li, Internet Multimedia Management Systems V, SPIE, Philadelphia, PA, USA, 2004, p. 157.
[13] A. Akhshani, S. Behnia, A. Akhavan, H.A. Hassan, Z. Hassan, Optics Communications 283 (2010) 3259.
[14] C. Fu, B.-b. Lin, Y.-s. Miao, X. Liu, J.-j. Chen, Optics Communications 284 (2011) 5415.
[15] C.K. Huang, H.H. Nien, Optics Communications 282 (2009) 2123.
[16] H. Kim, Y.H. Lee, Optics Communications 258 (2006) 9.
[17] H. Liu, X. Wang, Optics Communications 284 (2011) 3895.
[18] V. Patidar, N.K. Pareek, G. Purohit, K.K. Sud, Optics Communications 284 (2011) 4331.
[19] J. Zou, R.K. Ward, D. Qi, Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on, Vol. 963, 2004, p. III-965.
[20] J. Zou, R.K. Ward, D. Qi, Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on, vol. 383, 2004, p. iii-385.
[21] Y. Zhou, S. Agaian, V.M. Joyner, K. Panetta, IS&T / SPIE Electronic Imaging 2008: Image Processing: Algorithms and Systems VI, SPIE, San Jose, CA, 2008, p. 681215.
[22] Y. Zhou, K. Panetta, S. Agaian, SPIE Defense, Security, and Sensing 2008: Mobile Multimedia/Image Processing, Security, and Applications 2008, SPIE, Orlando, FL, 2008, 69820H–1.
[23] S. Li, C. Li, K.-T. Lo, G. Chen, IEEE Transactions on Circuits and Systems for Video Technology 18 (2008) 338.
[24] R.-J. Chen, S.-J. Horng, Signal Processing: Image Communication 25 (2010) 413.
[25] J.-W. Han, C.-S. Park, D.-H. Ryu, E.-S. Kim, Optical Engineering 38 (1999) 47.
[26] M. Podesser, H. Schmidt, A. Uhl, The 5th Nordic Signal Processing Symposium — NORSIG—2002, on board Hurtigruten, Norway, 2002, p. 1037.
[27] D. Moon, Y. Chung, S.B. Pan, K. Moon, K. Chung, ETRI Journal 28 (2006) 444.
[28] D. Engel, E. Pschernig, A. Uhl, IEEE Transactions on Information Forensics and Security 3 (2008) 173.
[29] D.Z. Gevorkian, K.O. Egiazarian, S.S. Agaian, J.T. Astola, O. Vainio, IEEE Transactions on Signal Processing 43 (1995) 286.
[30] S. Agaian, J. Astola, K. Egiazarian, P. Kuosmanen, Signal Processing 41 (1995) 101.
[31] D. Qi, J. Zou, X. Han, Science in China Series E: Technological Sciences 43 (2000) 304.
[32] C. Heuberger, Periodica Mathematica Hungarica 49 (2004) 65.
[33] S. Dey, A. Abraham, S. Sanyal, Information Assurance and Security, 2007. IAS 2007. Third International Symposium on, 2007, p. 101.
[34] R.C. Gonzalez, R.E. Woods, Digital Image Processing, 3 ed., Pearson Prentice Hall, 2007.
[35] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, IEEE Transactions on Image Processing 13 (2004) 600.
[36] ITSEC, Document COM(90) 314, Commission of the European Communities, 2006.
[37] G. Chen, Y. Mao, C.K. Chui, Chaos, Solitons & Fractals 21 (2004) 749.
[38] N.K. Pareek, V. Patidar, K.K. Sud, Image and Vision Computing 24 (2006) 926.
[39] Wikipedia, Correlation and dependence, in, http://en.wikipedia.org/wiki/Correlation_and_dependence.
[40] mathbits.com, Correlation Coefficient, in, http://mathbits.com/Mathbits/TISection/Statistics2/correlation.htm.
[41] B. Schneier, Applied Cryptography, Wiley, 1995.
[42] A.J. Menezes, P.C. Van Oorschot, S.A. Vanstone, Handbook of Applied Cryptography, CRC Press, Inc., New York, 1997.